arXiv:0906.4431v3 [cs.CC] 13 Nov 2009

The Complexity of Probabilistic Lobbying *

Gábor Erdélyi† and Henning Fernau‡ and Judy Goldsmith§
and
Nicholas Mattei¶ and Daniel Raible‖ and Jörg Rothe**

November 13, 2009

Abstract

We propose models for lobbying in a probabilistic environment, in which an actor (called "The Lobby") seeks to influence voters' preferences of voting for or against multiple issues when the voters' preferences are represented in terms of probabilities. In particular, we provide two evaluation criteria and three bribery methods to formally describe these models, and we consider the resulting forms of lobbying with and without issue weighting. We provide a formal analysis for these problems of lobbying in a stochastic environment, and determine their classical and parameterized complexity depending on the given bribery/evaluation criteria. Specifically, we show that some of these problems can be solved in polynomial time, some are NP-complete but fixed-parameter tractable, and some are W[2]-complete. Finally, we provide approximability and inapproximability results for these problems and several variants.

1 Introduction

In the American political system, laws are passed by elected officials who are supposed to represent their constituency. Individual entities such as citizens or corporations are not supposed to have

*A preliminary version of this paper appears in the proceedings of the *1st International Conference on Algorithmic Decision Theory*, October 2009 [8]. This work was supported in part by DFG grants RO 1202/11-1 and RO 1202/12-1, the European Science Foundation's EUROCORES program LogICCC, the Alexander von Humboldt Foundation's TransCoop program, and NSF grant ITR-0325063. This work was done in part while the first author was visiting Universität Trier, while the third author and the fifth author were visiting Heinrich-Heine-Universität Düsseldorf, and while the sixth author was visiting the University of Rochester.

†URL: ccc.cs.uni-duesseldorf.de/˜erdelyi. Heinrich-Heine-Universität Düsseldorf, Institut für Informatik, 40225 Düsseldorf, Germany.

‡URL: www.informatik.uni-trier.de/˜fernau. Universität Trier, FB 4—Abteilung Informatik, 54286 Trier, Germany.

§URL: www.cs.uky.edu/˜goldsmit. University of Kentucky, Dept. of Computer Science, Lexington, KY 40506, USA.

¶URL: www.cs.uky.edu/˜nsmatt2. University of Kentucky, Dept. of Computer Science, Lexington, KY 40506, USA.

‖URL: www.informatik.uni-trier.de/˜raible. Universität Trier, FB 4—Abteilung Informatik, 54286 Trier, Germany.

**URL: ccc.cs.uni-duesseldorf.de/˜rothe. Heinrich-Heine-Universität Düsseldorf, Institut für Informatik, 40225 Düsseldorf, Germany.

undue influence in the wording or passage of a law. However, they are allowed to make contributions to representatives, and it is common to include an indication that the contribution carries an expectation that the representative will vote a certain way on a particular issue.

Many factors can affect a representative's vote on a particular issue. There are the representative's personal beliefs about the issue, which presumably were part of the reason that the constituency elected them. There are also the campaign contributions, communications from constituents, communications from potential donors, and the representative's own expectations of further contributions and political support.

It is a complicated process to reason about. Earlier work considered the problem of meting out contributions to representatives in order to pass a set of laws or influence a set of votes. However, the earlier computational complexity work on this problem made the assumption that a politician who accepts a contribution will in fact—if the contribution meets a given threshold—vote according to the wishes of the donor.

It is said that "An honest politician is one who stays bought," but that does not take into account the ongoing pressures from personal convictions and opposing lobbyists and donors. We consider the problem of influencing a set of votes under the assumption that we can influence only the *probability* that the politician votes as we desire.

There are several axes along which we complicate the picture. The first is the notion of sufficiency: What does it mean to say we have donated enough to influence the vote? Does it mean that the probability that a single vote will go our way is greater than some threshold? That the probability that all the votes go our way is greater than that threshold? We discuss these and other criteria in the section on evaluation criteria.

How does one donate money to a campaign? In the United States there are several laws that influence how, when, and how much a particular person or organization can donate to a particular candidate. We examine ways in which money can be channeled into the political process in Section 2.

Lobbying has been studied formally by economists, computer scientists, and special interest groups since at least 1983 [20] and as an extension to formal game theory since 1944 [22]. The different disciplines have considered mostly disjoint aspects of the process while seeking to accomplish distinct goals with their respective formal models. Economists study lobbying as "economic games," as defined by von Neumann and Morgenstern [22]. This analysis is focused on learning how these complex systems work and deducing optimal strategies for winning the competitions [20,1,2]. This work has also focused on how to "rig" a vote and how to optimally dispense the funds among the various individuals [1]. Economists are interested in finding effective and efficient bribery schemes [1] as well as determining strategies for instances of two or more players [1,20, 2]. Generally, they reduce the problem of finding an effective lobbying strategy to one of finding a winning strategy for the specific type of game. Economists have also formalized this problem for bribery systems in both the United States [20] and the European Union [6].

In the emerging field of computational social choice, voting and preference aggregation are studied from a computational perspective, with a particular focus on the complexity of winner determination, manipulation, procedural control, and bribery in elections (see, e.g., the survey [13] and the references cited therein), and also with respect to lobbying in the context of direct democ-

racy where voters vote on multiple referenda. In particular, Christian et al. [5] show that "Optimal Lobbying" (OL) is complete for the (parameterized) complexity class W[2]. The OL problem is a deterministic and unweighted version of the problems that we present in this paper. Sandholm noted that the "Optimal Weighted Lobbying" (OWL) problem, which allows different voters to have different prices, can be expressed as and solved via the "binary multi-unit combinatorial reverse auction winner-determination problem" (see [21]).

We extend the models of lobbying, and provide algorithms and analysis for these extended models in terms of classical and parameterized complexity. We also provide approximability and inapproximability results. In this way we add breadth and depth to not only the models but also the understanding of lobbying behavior.

2 Models for Probabilistic Lobbying

2.1 Initial Model

We begin with a simplistic version of the PROBABILISTIC LOBBYING PROBLEM (PLP, for short), in which voters start with initial probabilities of voting for an issue and are assigned known costs for increasing their probabilities of voting according to "The Lobby's" agenda by each of a finite set of increments. The question, for this class of problems, is: Given the above information, along with an agenda and a fixed budget B, can The Lobby target its bribes in order to achieve its agenda?

The complexity of the problem seems to hinge on the evaluation criterion for what it means to "win a vote" or "achieve an agenda." We discuss the possible interpretations of evaluation and bribery later in this section.[1] First, however, we will formalize the problem by defining data objects needed to represent the problem instances. (A similar model was first discussed by Reinganum [20] in the continuous case and we translate it here to the discrete case. This will allow us to present algorithms for, and a complexity analysis of, the problem.)

Let $\mathbb{Q}_{[0,1]}^{m\times n}$ denote the set of $m\times n$ matrices over $\mathbb{Q}_{[0,1]}$ (the rational numbers in the interval $[0,1]$). We say $P \in \mathbb{Q}_{[0,1]}^{m\times n}$ is a probability matrix (of size $m\times n$), where each entry $p_{i,j}$ of P gives the probability that voter v_i will vote "yes" for referendum (synonymously, for issue) r_j. The result of a vote can be either a "yes" (represented by 1) or a "no" (represented by 0). Thus, we can represent the result of any vote on all issues as a 0/1 vector $\vec{X} = (x_1, x_2, \ldots, x_n)$, which is sometimes also denoted as a string in $\{0,1\}^n$.

We now associate with each voter/issue pair (v_i, r_j) a discrete price function $c_{i,j}$ for changing v_i's probability of voting "yes" for issue r_j. Intuitively, $c_{i,j}$ gives the cost for The Lobby of raising or lowering (in discrete steps) the ith voter's probability of voting "yes" on the jth issue. A formal description is as follows.

Given the entries $p_{i,j} = a_{i,j}/b_{i,j}$ of a probability matrix $P \in \mathbb{Q}_{[0,1]}^{m\times n}$, where $a_{i,j} \in \mathbb{N} = \{0,1,\ldots\}$ and $b_{i,j} \in \mathbb{N}_+ = \{1,2,\ldots\}$, choose some $k \in \mathbb{N}$ such that $k+1$ is a common multiple of all $b_{i,j}$, where $1 \leq i \leq m$ and $1 \leq j \leq n$, and partition the probability interval $[0,1]$ into $k+1$ steps of

[1] We stress that when we use the term "bribery" in this paper, it is meant in the sense of lobbying [5], not in the sense Faliszewski et al. [9] define bribery (see also, e.g., [10,11,12]).

size ${}^1\!/\!{\scriptstyle (k+1)}$ each.[2] The integer k will be called the discretization level of the problem. For each $i \in \{1,2,\ldots,m\}$ and $j \in \{1,2,\ldots,n\}$, $c_{i,j} : \{0, {}^1\!/\!{\scriptstyle (k+1)}, {}^2\!/\!{\scriptstyle (k+1)}, \ldots, {}^k\!/\!{\scriptstyle (k+1)}, 1\} \to \mathbb{N}$ is the *(discrete) price function for $p_{i,j}$*, i.e., $c_{i,j}({}^\ell\!/\!{\scriptstyle (k+1)})$ is the price for changing the probability of the ith voter voting "yes" on the jth issue from $p_{i,j}$ to ${}^\ell\!/\!{\scriptstyle (k+1)}$, where $0 \le \ell \le k+1$. Note that the domain of $c_{i,j}$ consists of $k+2$ elements of $\mathbb{Q}_{[0,1]}$ including 0, $p_{i,j}$, and 1. In particular, we require $c_{i,j}(p_{i,j}) = 0$, i.e., a cost of zero is associated with leaving the initial probability of voter v_i voting on issue r_j unchanged. Note that $k = 0$ means $p_{i,j} \in \{0,1\}$, i.e., in this case each voter either accepts or rejects each issue with certainty and The Lobby can only flip these results.[3] The image of $c_{i,j}$ consists of $k+2$ nonnegative integers including 0, and we require that, for any two elements a, b in the domain of $c_{i,j}$, if $p_{i,j} \le a \le b$ or $p_{i,j} \ge a \ge b$, then $c_{i,j}(a) \le c_{i,j}(b)$. This guarantees monotonicity on the prices.

We represent the list of price functions associated with a probability matrix P as a table C_P, called cost matrix in the following, whose $m \cdot n$ rows give the price functions $c_{i,j}$ and whose $k+2$ columns give the costs $c_{i,j}({}^\ell\!/\!{\scriptstyle (k+1)})$, where $0 \le \ell \le k+1$. Note that we choose the same k for each $c_{i,j}$, so we have the same number of columns in each row of C_P. The entries of C_P can be thought of as "price tags" indicating what The Lobby must pay in order to change the probabilities of voting.

The Lobby also has an integer-valued budget B and an "agenda," which we will denote as a vector $\vec{Z} \in \{0,1\}^n$ for n issues, containing the outcomes The Lobby would like to see on these issues. For The Lobby, the prices for a bribery that moves the outcomes of a referendum into the wrong direction do not matter. Hence, if $\vec{Z}$ is zero at position j, then we can set $c_{i,j}(a) = --$ (indicating an unimportant entry) for $a > p_{i,j}$, and if $\vec{Z}$ is one at position j, then we can set $c_{i,j}(a) = --$ (indicating an unimportant entry) for $a < p_{i,j}$. Without loss of generality, we can also assume that $c_{i,j}(a) = 0$ if and only if $a = p_{i,j}$.

For simplicity, we may occasionally assume that The Lobby's agenda is all "yes" votes, so the target vector is $\vec{Z} = 1^n$. This assumption can be made without loss of generality, since if there is a zero in $\vec{Z}$ at position j, we can flip this zero to one and also change the corresponding probabilities $p_{1,j}, p_{2,j}, \ldots, p_{m,j}$ in the jth column of P to $1-p_{1,j}, 1-p_{2,j}, \ldots, 1-p_{m,j}$. (See the evaluation criteria in Section 2.3 for how to determine the result of voting on a referendum.) Moreover, the rows of the cost matrix C_P that correspond to issue j have to be mirrored.

Example 1 *Consider the following problem instance with $k = 9$ (so there are $k+1 = 10$ steps), $m = 2$ voters, and $n = 3$ issues. We will use this as a running example for the rest of this paper. In addition to the above definitions for k, m, and n, we also give the following matrix for P. (Note that this example is normalized for an agenda of $\vec{Z} = 1^3$, which is why The Lobby has no incentive for lowering the acceptance probabilities, so those costs are omitted below.)*

[2]There is some arbitrariness in this choice of k. One might think of more flexible ways of partitioning $[0,1]$. We have chosen this way for the sake of simplifying the representation, but we mention that all that matters is that for each i and j, the discrete price function $c_{i,j}$ is defined on the value $p_{i,j}$, and is set to zero for this value.

[3]This is the special case of Optimal Lobbying.

Our example consists of a probability matrix P:

	r_1	r_2	r_3
v_1	0.8	0.3	0.5
v_2	0.4	0.7	0.4

and the corresponding cost matrix C_P:

$c_{i,j}$	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$c_{1,1}$	−−	−−	−−	−−	−−	−−	−−	−−	0	100	140
$c_{1,2}$	−−	−−	−−	0	10	70	100	140	310	520	600
$c_{1,3}$	−−	−−	−−	−−	−−	0	15	25	70	90	150
$c_{2,1}$	−−	−−	−−	−−	0	30	40	70	120	200	270
$c_{2,2}$	−−	−−	−−	−−	−−	−−	−−	0	10	40	90
$c_{2,3}$	−−	−−	−−	−−	0	70	90	100	180	300	450

In Section 2.2, we describe three bribery methods, i.e., three specific ways in which The Lobby can influence the voters. These will be referred to as *microbribery* (MB), *issue bribery* (IB), and *voter bribery* (VB). In addition to the three bribery methods described in Section 2.2, we also define two ways in which The Lobby can win a set of votes. These evaluation criteria are defined in Section 2.3 and will be referred to as *strict majority* (SM) and *average majority* (AM).

We now introduce the six basic problems that we will study. For $\mathrm{X} \in \{\mathrm{MB}, \mathrm{IB}, \mathrm{VB}\}$ and $\mathrm{Y} \in \{\mathrm{SM}, \mathrm{AM}\}$, we define the following problem.

Name: X-Y PROBABILISTIC LOBBYING PROBLEM.

Given: A probability matrix $P \in \mathbb{Q}_{[0,1]}^{m \times n}$ with a cost matrix C_P (with integer entries), a target vector $\vec{Z} \in \{0,1\}^n$, and a budget B.

Question: Is there a way for The Lobby to influence P (using bribery method X and evaluation criterion Y, without exceeding budget B) such that the result of the votes on all issues equals $\vec{Z}$?

We abbreviate this problem name as X-Y-PLP.

Occasionally, we will study further variants of the basic problems that will be formally described below.

2.2 Bribery Methods

We begin by first formalizing the bribery methods by which The Lobby can influence votes on issues. We will define three methods for donating this money.

2.2.1 Microbribery (MB)

The first method at the disposal of The Lobby is what we will call *microbribery*.[4] We define microbribery to be the editing of individual elements of the P matrix according to the costs in the C_P matrix. Thus The Lobby picks not only which voter to influence but also which issue to influence for that voter. This bribery method allows the most flexible version of bribery, and models private donations made to candidates in support of specific issues.

2.2.2 Issue Bribery (IB)

The second method at the disposal of The Lobby is *issue bribery*. We can see from the P matrix that each column represents how the voters think about a particular issue. In this method of bribery, The Lobby can pick a column of the matrix and edit it according to some budget. The money will be equally distributed among all the voters and the voter probabilities will move accordingly. So, for d dollars each voter receives a fraction of d/m and his or her probability of voting "yes" changes accordingly. This can be thought of as special-interest group donations. Special-interest groups such as PETA[5] focus on issues and dispense their funds across an issue rather than by voter. The bribery could be funneled through such groups.

2.2.3 Voter Bribery (VB)

The third and final method at the disposal of The Lobby is *voter bribery*. We can see from the P matrix that each row represents what an individual voter thinks about all the issues on the docket. In this method of bribery, The Lobby picks a voter and then pays to edit the entire row at once with the funds being equally distributed over all the issues. So, for d dollars a fraction of d/n is spent on each issue, which moves accordingly. The cost of moving the voter is given by the C_P matrix as before. This method of bribery is analogous to "buying" or pushing a single politician or voter. The Lobby seeks to donate so much money to some individual voters that they have no choice but to move all of their votes toward The Lobby's agenda.

Simple Observations

Note that microbribery is equivalent to issue bribery if there is only one voter. Similarly, microbribery is equivalent to voter bribery if there is only one referendum.

2.3 Evaluation Criteria

Defining criteria for how an issue is won is the next important step in formalizing our models. Here we define two methods that one could use to evaluate the eventual outcome of a vote. Since we

[4] Although our notion was inspired by theirs, we stress that it should not be confused with the term "microbribery" used by Faliszewski et al. [10,11,12] in the different context of bribing "irrational" voters in Llull/Copeland elections via flipping single entries in their preference tables.

[5] People for the Ethical Treatment of Animals, a narrow-focus group that protests animal testing of food and drugs, and the swatting of flies.

are focusing on problems that are probabilistic in nature, it is important to note that no evaluation criterion will guarantee a win. The criteria below yield different outcomes depending on the model and problem instance.

2.3.1 Strict Majority (SM)

For each issue, a strict majority of the individual voters have probability greater than some threshold, t, of voting according to the agenda. In our running example (see Example 1), with $t = 50\%$, the result of the votes would be $\vec{X} = (0,0,0)$, because none of the issues has a strict majority of voters with above 50% likelihood of voting "yes."

2.3.2 Average Majority (AM)

For each issue r_j of a given probability matrix P, we define the average probability $\overline{p_j} = (\Sigma_{i=1}^{m} p_{i,j})/m$ of voting "yes" for r_j. We can now evaluate the vote to say that r_j is accepted if and only if $\overline{p_j} > t$ where t is some threshold. In our running example with $t = 50\%$, this would give us a result vector of $\vec{X} = (1,0,0)$.

Simple Observations

Note that both criteria coincide if there is only one voter or if the discretization level equals zero.

2.4 Issue Weighting

Our modification to the model will bring in the concept of issue weighting. It is reasonable to surmise that certain issues will be of more importance to The Lobby than others. For this reason we will allow The Lobby to specify higher weights to the issues that they deem more important. These positive integer weights will be defined for each issue.

We will specify these weights as a vector $\vec{W} \in \mathbb{N}_{+}^{n}$ with size n equal to the total number of issues in our problem instance. The higher the weight, the more important that particular issue is to The Lobby. Along with the weights for each issue we are also given an objective value $O \in \mathbb{N}_{+}$, which is the minimum weight The Lobby wants to see passed. Since this is a partial ordering, it is possible for The Lobby to have an ordering such as $w_1 = w_2 = \cdots = w_n$. If this is the case, we see that we are left with an instance of X-Y-PLP, where $\mathrm{X} \in \{\mathrm{MB}, \mathrm{IB}, \mathrm{VB}\}$ and $\mathrm{Y} \in \{\mathrm{SM}, \mathrm{AM}\}$.

We now introduce the six probabilistic lobbying problems with issue weighting. For $\mathrm{X} \in \{\mathrm{MB}, \mathrm{IB}, \mathrm{VB}\}$ and $\mathrm{Y} \in \{\mathrm{SM}, \mathrm{AM}\}$, we define the following problem.

Name: X-Y PROBABILISTIC LOBBYING PROBLEM WITH ISSUE WEIGHTING.

Given: A probability matrix $P \in \mathbb{Q}_{[0,1]}^{m \times n}$ with cost matrix C_P and a lobby target vector $\vec{Z} \in \{0,1\}^n$, a lobby weight vector $\vec{W} \in \mathbb{N}_{+}^{n}$, an objective value $O \in \mathbb{N}_{+}$, and a budget B.

Question: Is there a way for The Lobby to influence P (using bribery method X and evaluation criterion Y, without exceeding budget B) such that the total weight of all issues for which the result coincides with The Lobby's target vector $\vec{Z}$ is at least O?

We abbreviate this problem name as X-Y-PLP-WIW.

2.5 Exact Microbribery

The above definitions give rise to the natural question of exact bribery problems. The exact variants of probabilistic lobbying via microbribery, denoted by EXACT-MB-Y-PLP with $\mathrm{Y} \in \{\mathrm{SM}, \mathrm{AM}\}$, ask whether The Lobby can achieve its goal via microbribery (for the given evaluation criterion, SM or AM) by spending exactly B dollars. Thus, in these variants The Lobby is constrained to spending a total amount of exactly B dollars, no more, no less.

3 Complexity-Theoretic Notions

We assume the reader is familiar with standard notions of (classical) complexity theory, such as P, NP, and NP-completeness. Since we analyze the problems stated in Section 2 not only in terms of their classical complexity, but also with regard to their *parameterized* complexity, we provide some basic notions here (see, e.g., [7,14,18] for more background). As we derive our results in a rather specific fashion, we will employ the "Turing way" as proposed by Cesati [4].

A *parameterized problem* $\mathscr{P}$ is a subset of $\Sigma^* \times \mathbb{N}$, where Σ is a fixed alphabet and Σ^* is the set of strings over Σ. Each instance of the parameterized problem $\mathscr{P}$ is a pair (I,k), where the second component k is called the *parameter*. The language $L(\mathscr{P})$ is the set of all YES instances of $\mathscr{P}$. The parameterized problem $\mathscr{P}$ is *fixed-parameter tractable* if there is an algorithm (realizable by a deterministic Turing machine) that decides whether an input (I,k) is a member of $L(\mathscr{P})$ in time $f(k)|I|^c$, where c is a fixed constant and f is a function of the parameter, k, but is independent of the overall input length, $|I|$. The class of all fixed-parameter tractable problems is denoted by FPT.

The $\mathscr{O}^*(\cdot)$ notation has by now become standard in exact algorithms. It neglects not only constants (as the more familiar $\mathscr{O}(\cdot)$ notation does) but also polynomial factors in the function estimates. Thus, a problem is in FPT if and only if an instance (with parameter k) can be solved in time $\mathscr{O}^*(f(k))$ for some function f.

Sometimes, more than one parameter (e.g., two parameters (k_1, k_2)) can be associated with a (classical) problem. This can be formally captured in the definition above by coding those parameters into one number k via a so-called pairing function through diagonalization.

There is also a theory of parameterized complexity, as exhibited in [7,14,18], where parameterized complexity is expressed via hardness for or completeness in the levels W[t], $t \geq 1$, of the W-hierarchy, which complement fixed-parameter tractability:

$$\mathrm{FPT} = \mathrm{W}[0] \subseteq \mathrm{W}[1] \subseteq \mathrm{W}[2] \subseteq \cdots.$$

It is commonly believed that this hierarchy is strict. Since only the second level, W[2], will be of interest to us in this paper, we will define only this class below.

A *parameterized reduction* is a function r that, for some polynomial p and some function g, is computable in time $\mathscr{O}(g(k)p(|I|))$ and maps an instance (I,k) of $\mathscr{P}$ onto an instance $r(I,k) = (I',k')$ of $\mathscr{P}'$ such that

1. (I,k) is a YES instance of $\mathscr{P}$ if and only if (I',k') is a YES instance of $\mathscr{P}'$, and

2. $k' \leq g(k)$.

We then say that $\mathscr{P}$ *parameterized reduces to* $\mathscr{P}'$. Parameterized hardness for and completeness in a parameterized complexity class is defined via parameterized reductions. We will show only W[2]-completeness results. A parameterized problem $\mathscr{P}'$ is said to be W[2]-*hard* if every parameterized problem $\mathscr{P}$ in W[2] parameterized reduces to $\mathscr{P}'$. $\mathscr{P}'$ is said to be W[2]-*complete* if $\mathscr{P}'$ is in W[2] and is W[2]-hard.

W[2] can be characterized by the following problem on Turing machines:

Name: SHORT MULTI-TAPE NONDETERMINISTIC TURING MACHINE COMPUTATION.

Given: A multi-tape nondeterministic Turing machine M (with two-way infinite tapes) and an input string x (both M and x are given in some standard encoding).

Parameter: A positive integer k.

Question: Is there an accepting computation of M on input x that reaches a final accepting state in at most k steps?

We abbreviate this problem name as SMNTMC.

More specifically, a parameterized problem $\mathscr{P}$ is in W[2] if and only if it can be reduced to SMNTMC via a parameterized reduction [4]. This can be accomplished by giving an appropriate multi-tape nondeterministic Turing machine for solving $\mathscr{P}$. Hardness for W[2] can be shown by giving a parameterized reduction in the opposite direction, from SMNTMC to $\mathscr{P}$.

The parameterized complexity of a problem depends on the chosen parameterization. For classical problems that involve a budget $B \in \mathbb{N}$ (and hence can be viewed as minimization problems), the most straightforward parameterization is the given budget bound B. In this sense, we mostly state parameterized results in this paper, although sometimes different parameterizations are considered. (For other applications of fixed-parameter tractability and parameterized complexity to problems from computational social choice, see, e.g., [17].)

4 Classical Complexity Results

We now provide a formal complexity analysis of the probabilistic lobbying problems for all combinations of evaluation criteria and bribery methods.

Table 1 summarizes our results for X-Y-PLP, where $\mathrm{X} \in \{\mathrm{MB}, \mathrm{IB}, \mathrm{VB}\}$ and $\mathrm{Y} \in \{\mathrm{SM}, \mathrm{AM}\}$. Some of these results are known from previous work by Christian et al. [5], as will be mentioned below. Our results generalize theirs by extending the model to probabilistic settings. The listed FPT results might look peculiar at first glance, since Christian et al. [5] derived W[2]-hardness results, but this is due to the chosen parameterization, as will be discussed later in more detail. We put parentheses around some classes to indicate that these results are trivially inherited from others. For example, if some problem is solvable in polynomial time, then it is in FPT for any parameterization. The table mainly provides results on the containment of problems in certain complexity classes; if known, additional hardness results are also listed.

Problem	Classical Complexity	Parameterized Complexity, parameterized by		
		Budget	Budget per Issue	Budget & Discretiz. Level
MB-SM-PLP	P	(FPT)	(FPT)	(FPT)
MB-AM-PLP	P	(FPT)	(FPT)	(FPT)
IB-SM-PLP	P	(FPT)	(FPT)	(FPT)
IB-AM-PLP	P	(FPT)	(FPT)	(FPT)
VB-SM-PLP	NP-complete	FPT	W[2]-complete	(FPT)
VB-AM-PLP	NP-complete	W[2]	W[2]-complete	FPT

Table 1: Complexity results for X-Y-PLP, where X $\in$ {MB, IB, VB} and Y $\in$ {SM, AM}

4.1 Microbribery

Theorem 2 MB-SM-PLP *is in* P.

Proof. The aim is to win all referenda. For each voter v_i and referendum r_j, $1 \leq i \leq m$ and $1 \leq j \leq n$, we can compute in polynomial time the amount $b(v_i, r_j)$ The Lobby has to spend to turn the favor of v_i in the direction of The Lobby (beyond the given threshold t). In particular, set $b(v_i, r_j) = 0$ if voter v_i would already vote according to the agenda of The Lobby. For each issue r_j, sort $\{b(v_i, r_j) \mid 1 \leq i \leq m\}$ non-decreasingly, yielding a sequence $b_1(r_j), \ldots, b_m(r_j)$ such that $b_k(r_j) \leq b_\ell(r_j)$ for $k < \ell$. To win referendum r_j, The Lobby must spend at least $B(r_j) = \sum_{i=1}^{\lceil (m+1)/2 \rceil} b_i(r_j)$ dollars. Hence, all referenda can be won if and only if $\sum_{j=1}^{n} B(r_j)$ is at most the given bribery budget B. ❑

Note that the time needed to implement the algorithm given in the previous proof can be bounded by a polynomial of low order. More precisely, if the input consists of m voters, n referenda, and discretization level k, then $\mathcal{O}(n \cdot m \cdot k)$ time is needed to compute the $b(v_i, r_j)$. Having these values, $\mathcal{O}(n \cdot m \cdot \log(m))$ time is needed for the sorting phase. The sums can be computed in time $\mathcal{O}(n \cdot m)$.

Similarly, the other problems that we show to belong to P admit solution algorithms bounded by polynomials of low order.

The complexity of microbribery with evaluation criterion AM is somewhat harder to determine. We use the following auxiliary problem. Given a directed graph G consisting of path components $P_1, \ldots, P_\pi$ with vertex set $V = \{J_1, \ldots, J_n\}$, a *schedule* S of $q \leq n$ jobs (on a single machine) is a sequence $J_{i,1}, \ldots, J_{i,q}$ such that $J_{i(r)} = J_{i(s)}$ implies $r = s$. Assigning cost $c(J_k)$ to job J_k for each k, $1 \leq k \leq n$, the *cost of schedule* S is $c(S) = \sum_{k=1}^{q} c(J_{i(k)})$. S is said to *respect the precedence constraints of* G if for every path-component $P_i = (J_{i,1}, \ldots, J_{i,p(i)})$ of G (with $V = \bigcup_{i=1}^{\pi} \{J_{i,\ell} \mid 1 \leq \ell \leq p(i)\}$) and for each ℓ with $2 \leq \ell \leq p(i)$, we have: If $J_{i,\ell}$ occurs in schedule S then $J_{i,\ell-1}$ occurs in S before $J_{i,\ell}$.

Name: PATH SCHEDULE

Given: A set $V = \{J_1, \ldots, J_n\}$ of jobs, a directed graph $G = (V, A)$ consisting of pairwise disjoint paths $P_1, \ldots, P_z$, two numbers $C, q \in \mathbb{N}$, and a cost function $c : V \to \mathbb{N}$.

Question: Can we find a schedule $J_{i(1)}, \dots, J_{i(q)}$ of q jobs of cost at most C respecting the precedence constraints of G?

We first show, as Lemma 3, that PATH SCHEDULE is in P. Then we will show, as Theorem 4, how to reduce MB-AM-PLP to PATH SCHEDULE, which implies that MB-AM-PLP is in P as well.

Lemma 3 PATH SCHEDULE *is in* P.

Proof. Given an instance of PATH SCHEDULE as in the problem description above, the following dynamic programming approach calculates $T[\{P_1, \dots, P_z\}, q]$, which gives the minimum cost to solve the problem. We build up a table $T[\{P_1, \dots, P_\ell\}, j]$ storing the minimum cost of scheduling j jobs out of the jobs contained in the paths $P_1, \dots, P_\ell$. Let $P_i = J_{i,1}, \dots, J_{i,p(i)}$ be a path, $1 \leq i \leq z$. For $k \leq p(1)$, set $T[\{P_1\}, k] = \sum_{s=1}^{k} c(J_{1,s})$. For $k > p(1)$, set $T[\{P_1\}, k] = \infty$. If $\ell > 1$, $T[\{P_1, \dots, P_\ell\}, j]$ equals

$$\min_{0 \leq k \leq \min\{j, p(\ell)\}} T[\{P_1, \dots, P_{\ell-1}\}, j-k] + \sum_{s=1}^{k} c(J_{\ell,s}).$$

Consider each possible scheduling of the first k jobs of P_ℓ. For the remaining $j-k$ jobs, look up a solution in the table. Note that we can re-order each schedule S so that all jobs from one path contiguously appear in S, without violating the precedence constraints by this re-ordering nor changing the cost of the schedule. Hence, $T[\{P_1, \dots, P_z\}, q]$ gives the minimum schedule cost. The number of entries in the table is $z \cdot q$, and computing each entry $T[\{P_1, \dots, P_\ell\}, \cdot]$ is linear in $p(\ell)$ (for each ℓ, $1 \leq \ell \leq z$), which leads to a run time of the dynamic programming algorithm that is polynomially bounded in the input size. ❑

Theorem 4 MB-AM-PLP *is in* P.

Proof. Let $(P, C_P, \vec{Z}, B)$ be a given MB-AM-PLP instance, where $P \in \mathbb{Q}_{[0,1]}^{m \times n}$, C_P is a cost matrix, $\vec{Z} \in \{0,1\}^n$ is The Lobby's target vector, and B is its budget. Let k be the discretization level of P, i.e., the interval is divided into $k+1$ steps of size $1/(k+1)$ each. For $j \in \{1, 2, \dots, n\}$, let d_j be the minimum cost for The Lobby to bring referendum r_j into line with the jth entry of its target vector $\vec{Z}$. If $\sum_{j=1}^{n} d_j \leq B$, then The Lobby can achieve its goal that the votes on all issues equal $\vec{Z}$.

For simplicity, we assume in the following argument that $\vec{Z} = 1^n$ (see Section 2.1). For every r_j, create an equivalent PATH SCHEDULING instance. First, compute for r_j the minimum number b_j of bribery steps needed to achieve The Lobby's goal on r_j. That is, choose the smallest $b_j \in \mathbb{N}$ such that $\overline{p_j} + b_j/(k+1)m > t$, where t is the threshold mentioned in our evaluation criterion AM. Now, given r_j, derive a path P_i from the price function $c_{i,j}$ for every voter v_i, $1 \leq i \leq m$, as follows:

1. Let s, $0 \leq s \leq k+1$, be minimum with the property $c_{i,j}(s/(k+1)) \in \mathbb{N}_+$.
2. Create a path $P_i = ((p_s, i), \dots, (p_{k+1}, i))$, where $p_h = h/(k+1)$.
3. Assign the cost $\hat{c}((p_h, i)) = c_{i,j}(p_h) - c_{i,j}(p_{(h-1)})$ to (p_h, i), $s+1 \leq h \leq k+1$.

Note that $\hat{c}((p_h, i))$ represents the cost of raising the probability of voting "yes" from ${}^{(h-1)}/_{(k+1)}$ to ${}^{h}/_{(k+1)}$. In order to do so, we must have reached an acceptance probability of ${}^{(h-1)}/_{(k+1)}$ first. Now, let the number of jobs to be scheduled be b_j. Note that one can take b_j bribery steps at the cost of d_j dollars if and only if one can schedule b_j jobs with a cost of d_j. Hence, we can decide whether or not $(P, C_P, \vec{Z}, B)$ is in MB-AM-PLP by using the dynamic program given in the proof of Lemma 3. ❑

Exact Version of Microbribery

Recall the definitions of EXACT-MB-SM-PLP and EXACT-MB-AM-PLP from Section 2.5.

Theorem 5 EXACT-MB-SM-PLP *and* EXACT-MB-AM-PLP *are* NP-*complete.*

Proof. We focus on EXACT-MB-SM-PLP and note that the reduction can be carried over straightforwardly to the case of EXACT-MB-AM-PLP.

To see membership in NP, observe that an instance I of EXACT-MB-SM-PLP can be transformed into another instance I' of EXACT-MB-SM-PLP after guessing which issue and which voter should be bribed by an amount of money specified by the first non-zero entry in the corresponding row of the cost matrix C_P and modifying C_P accordingly. After repeated guesses, we either arrive at an amount of left-over money that cannot be used for any bribery anymore (since it is either too small or possibly no money can be spent at all, since all issues and voters have been "completely bribed"); in this case, the nondeterministic procedure simply rejects and stops. Or, all money was (exactly) spent. In that case, it is checked if the evaluation criterion was met. If it is met, the algorithm accepts and stops; otherwise, it rejects and stops. Hence, the nondeterministic procedure will either succeed in one branch, yielding an affirmative answer, or there is no solution, and no affirmative answer will be produced.

To show NP-hardness of EXACT-MB-SM-PLP, we provide a reduction from SUBSET SUM (see, e.g., Garey and Johnson [15]): Given $a_1, \ldots, a_n, S \in \mathbb{N}_+$, does there exist a subset $I \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in I} a_i = S$? For such a SUBSET SUM instance, create an EXACT-MB-SM-PLP instance with only one referendum r_1 and with voters $v_1, \ldots, v_{2n}$. Let $k = 0$ (so we have $p_{i,j} \in \{0, 1\}$; see Section 2.1). Set $P_{i,1} = 1$ and $c_{i,1}(1) = 0$ for $1 \leq i \leq n$, and set $P_{i,1} = 0$, $c_{i,1}(0) = 0$, $c_{i,1}(1) = a_{i-n}$ for $n+1 \leq i \leq 2n$. Let $B = S$ and $t = 0.5$. Observe that we have to influence at least one of the voters not in accordance with r_1. Thus we can turn r_1 to The Lobby's favor by spending exactly B dollars on a set of voters $v_{i_1}, \ldots, v_{i_\ell}$ ($n+1 \leq i_j \leq 2n$) if and only if there is a subset $I \subseteq \{1, \ldots, n\}$, $|I| = \ell$, such that $\sum_{i \in I} a_i = S$.

NP-hardness of EXACT-MB-AM-PLP follows via the same reduction, taking $t = 0.5$ as a threshold. ❑

4.2 Issue Bribery

Theorem 6 IB-SM-PLP *and* IB-AM-PLP *are in* P.

Proof. We prove that IB-SM-PLP is in P; the proof for IB-AM-PLP is analogous. Observe that IB-SM-PLP (just like the problem OL to be defined in Section 4.3 below) contains a vector that represents the issues that The Lobby would like to see passed. In Theorem 7 (see Section 4.3), the constraint from OL, b, is expressed over the number of *voters* that need to be influenced. In IB-SM-PLP, however, we are required to influence *issues*. Thus, in order to determine a win we construct a *cost difference* vector representing how much it would cost The Lobby to win each issue (since all voters receive the same amount of money, this can be determined in time polynomial in the number of voters and issues). If (and only if) the sum of these costs does not exceed the given budget, this is a "yes" instance. ❑

4.3 Voter Bribery

Christian et al. [5] proved that the following problem is W[2]-complete. We state this problem in the standard format for parameterized complexity:

Name: OPTIMAL LOBBYING.

Given: An $m \times n$ matrix E and a $0/1$ vector $\vec{Z}$ of length n. Each row of E represents a voter and each column represents an issue. $\vec{Z}$ represents The Lobby's target outcome.

Parameter: A positive integer b (representing the number of voters to be influenced).

Question: Is there a choice of b rows of the matrix (i.e., of b voters) that can be changed such that in each column of the resulting matrix (i.e., for each issue) a majority vote yields the outcome targeted by The Lobby?

We abbreviate this problem name as OL.

Christian et al. [5] proved that this problem is W[2]-complete by a reduction from k-DOMINATING SET to OL (showing the lower bound) and from OL to INDEPENDENT-k-DOMINATING SET (showing the upper bound). In particular, this implies NP-hardness of OL. The following result focuses on the classical complexity of VB-SM-PLP and VB-AM-PLP; the parameterized complexity of these problems will be studied in Section 5 and will make use of the proof of Theorem 7 below.

To employ Christian et al.'s W[2]-hardness result [5], we show that OL is a special case of VB-SM-PLP and thus (parameterized) polynomial-time reduces to VB-SM-PLP. Analogous arguments apply to VB-AM-PLP.

Theorem 7 VB-SM-PLP *and* VB-AM-PLP *are* NP-*complete.*

Proof. Membership in NP is easy to see for both VB-SM-PLP and VB-AM-PLP.

We now prove that VB-SM-PLP is NP-hard by reducing OL to VB-SM-PLP. We are given an instance $(E, \vec{Z}, b)$ of OL, where E is a $m \times n$ $0/1$ matrix, b is the number of votes to be edited, and $\vec{Z}$ is the agenda for The Lobby. Without loss of generality, we may assume that $\vec{Z} = 1^n$ (see Section 2.1).

Problem	Classical Complexity	Parameterized Complexity, parameterized by		
		Budget	Budget per Issue	Budget & Discretiz. Level
MB-SM-PLP-WIW	NP-complete	FPT	?	(FPT)
MB-AM-PLP-WIW	NP-complete	FPT	?	(FPT)
IB-SM-PLP-WIW	NP-complete	FPT	?	(FPT)
IB-AM-PLP-WIW	NP-complete	FPT	?	(FPT)
VB-SM-PLP-WIW	NP-complete	FPT	W[2]-complete*	(FPT)
VB-AM-PLP-WIW	NP-complete	W[2]*	W[2]-complete*	FPT

Table 2: Complexity results for X-Y-PLP-WIW, where $\mathrm{X} \in \{\mathrm{MB}, \mathrm{IB}, \mathrm{VB}\}$ and $\mathrm{Y} \in \{\mathrm{SM}, \mathrm{AM}\}$

We construct an instance of VB-SM-PLP consisting of the given matrix $P = E$ (a "degenerate" probability matrix with only the probabilities 0 and 1), a corresponding cost matrix C_P, a target vector $\vec{Z} = 1^n$, and a budget B. C_P has two columns (i.e., we have $k = 0$, since the problem instance is deterministic, see Section 2.1), one column for probability 0 and one for probability 1. All entries of C_P are set to unit cost.

The cost of increasing any value in P is n, since donations are distributed evenly across issues for a given voter. We want to know whether there is a set of bribes of cost at most $b \cdot n = B$ such that The Lobby's agenda passes. This holds if and only if there are b voters that can be bribed so that they vote uniformly according to The Lobby's agenda and that is sufficient to pass all the issues. Thus, the given instance $(E, \vec{Z}, b)$ is in OL if and only if the constructed instance $(P, C_P, \vec{Z}, B)$ is in VB-SM-PLP, which shows that OL is a polynomial-time recognizable special case of VB-SM-PLP, and thus VB-SM-PLP is NP-hard.

Note that for the construction above it does not matter whether we use the strict-majority criterion (SM) or the average-majority criterion (AM). Since the entries of P are 0 or 1, we have $\overline{p_j} > 0.5$ if and only if we have a strict majority of ones in the jth column. Thus, VB-AM-PLP is NP-hard too. ❑

4.4 Probabilistic Lobbying with Issue Weighting

Table 2 summarizes our results for X-Y-PLP-WIW, where $\mathrm{X} \in \{\mathrm{MB}, \mathrm{IB}, \mathrm{VB}\}$ and $\mathrm{Y} \in \{\mathrm{SM}, \mathrm{AM}\}$. The most interesting observation is that introducing issue weights raises the complexity from P to NP-completeness for all cases of microbribery and issue bribery (though it remains the same for voter bribery). Nonetheless, we show (Theorem 15) that these NP-complete problems are fixed-parameter tractable. Another interesting observation concerns the question of membership in W[2]. In the cases indicated by the $*$ annotation, we can show this membership only when we take the lower bound O quantifying the objective of the bribery (in terms of issue weights) as a further parameter. Question marks indicate open problems.

Theorem 8 *The problems* MB-SM-PLP-WIW, MB-AM-PLP-WIW, IB-SM-PLP-WIW, *and* IB-AM-PLP-WIW *are* NP-*complete.*

Proof. Membership in NP is easy to see for each problem X-Y-PLP-WIW, $X \in \{MB, IB\}$ and $Y \in \{SM, AM\}$.

To prove that MB-SM-PLP-WIW is NP-hard, we give a reduction from the well-known NP-complete problem KNAPSACK (see, e.g., [15]) to MB-SM-PLP-WIW. In KNAPSACK, we are given a set of objects $U = \{o_1, \ldots, o_n\}$ with weights $w : U \to \mathbb{N}$ and profits $p : U \to \mathbb{N}$, and $W, P \in \mathbb{N}$. The question is whether there is a subset $I \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in I} w(o_i) \leq W$ and $\sum_{i \in I} p(o_i) \geq P$. Given a KNAPSACK instance (U, w, p, W, P), create a MB-SM-PLP-WIW instance with $k = 0$ and only one voter, v_1, where for each issue, v_1's acceptance probability is either zero or one. For each object $o_j \in U$, create an issue r_j such that the acceptance probability of v_1 is zero. Let the cost of raising this probability on r_j be $c_{1,j}(1) = w(o_j)$ and let the weight of issue r_j be $w_j = p(o_j)$. Let The Lobby's budget be W and its objective value be $O = P$. By construction, there is a subset $I \subseteq \{1, \ldots, n\}$ with $\sum_{i \in I} w(o_i) \leq W$ and $\sum_{i \in I} p(o_i) \geq P$ if and only if there is a subset $I \subseteq \{1, \ldots, n\}$ with $\sum_{i \in I} c_{1,i} \leq W$ and $\sum_{i \in I} w_i \geq O$.

As the reduction introduces only one voter, there is no difference between the bribery methods MB and IB, and no difference either between the evaluation criteria SM and AM. Hence, the above reduction works for all four problems. ❑

Turning now to voter bribery with issue weighting, note that an immediate consequence of Theorem 7 is that VB-SM-PLP-WIW and VB-AM-PLP-WIW are NP-hard, since they are generalizations of VB-SM-PLP and VB-AM-PLP. Again, membership in NP is also easy to see for the more general problems.

Corollary 9 VB-SM-PLP-WIW *and* VB-AM-PLP-WIW *are* NP-*complete.*

5 Parameterized Complexity Results

In this section, we study the parameterized complexity of our probabilistic lobbying problems. Parameterized hardness is usually shown by proving hardness for the levels of the W-hierarchy (with respect to parameterized reductions). Indeed, this hierarchy may be viewed as a "barometer of parametric intractability" [7, p. 14]. The lowest two levels of the W-hierarchy, $W[0] = FPT$ and $W[1]$, are the parameterized analogues of the classical complexity classes P and NP. We will show completeness results for the $W[2]$ level of this hierarchy.

5.1 Voter Bribery

Theorem 10 VB-SM-PLP *and* VB-AM-PLP *(parameterized by the budget and by the discretization level) is in* FPT.

Proof. Consider an instance of VB-Y-PLP, $Y \in \{SM, AM\}$, i.e., we are given n referenda and m voters, as well as a cost matrix C_P (with $--$ or integer entries), a discretization level k, and a budget limit B. We can assume that the target $\vec{Z}$ of The Lobby is 1^n. Hence, the rows of C_P are monotonically non-decreasing (after some $--$ entries). Observe that any successful bribe of any voter needs at least n dollars, since the money is evenly distributed among all referenda, and at

least one dollar is needed to influence any referendum vote of the chosen voter. Hence, $B \geq n$. We can assume that any entry in C_P is limited by B, since there would not be enough money in the budget to perform that kind of bribery. Although k could be bigger than B, the interesting area of each row in C_P (containing integer entries) cannot have more than B strict increases in the sequence. We can therefore encode each row in C_P by a sequence $(k_1, b_1, k_2, b_2, \ldots, k_\ell, b_\ell)$, $\ell < B$, which reads as follows: By investing b_j dollars, we can proceed to column number $\sum_{i \leq j} k_i$. Note that k is given in unary in the original instance (implicitly by giving the cost matrix C_P), so that each k_j can be encoded with $\log(k)$ bits. Hence, we can extract from C_P for each voter v a submatrix $S_P(v)$ with $n \leq B$ rows (for the referenda) and at most $2B$ columns (encoding the "jumps" in the integer sequence as described above). As already mentioned, the integer entries of C_P are bounded by B, so we can associate with each voter at most $(B + \log(k))^{B^2}$ distinct submatrices $S_P(v)$ of this kind, called voter profiles in the following. It makes no sense to store more than B voters with the same profile. Hence, we can assume that $m \leq B \cdot (B + \log(k) + 1)^{B^2}$. Therefore, all relevant parts of the input are bounded by a function in the parameters B and k,[6] so that some brute-force algorithm can be used to solve the instance. This shows that the problem is in FPT. For example, a simple search-tree algorithm would take, as long as possible, n dollars off the budget and then test all m possibilities of spending the money, modifying the voter profiles throughout the branching process. ❑

In practice, it is reasonable to assume that the discretization level is a rather small number. Although we were not able to establish an FPT-result for VB-AM-PLP, assuming that the discretization level is not part of the parameter, we can overcome this formal obstacle for VB-SM-PLP, as the following result shows.

Theorem 11 VB-SM-PLP *(parameterized by the budget) is in* FPT.

Proof. From the given cost matrix C_P, we can extract the information $W(i, j)$ that gives the minimum amount of money The Lobby must spend on voter v_i to turn his or her voting behavior on issue r_j in favor of The Lobby's agenda, eventually raising the corresponding voting probability beyond the given threshold t. Each entry in $W(i, j)$ is between 0 and B. Moreover, as argued in the previous proof, there are no more than B issues and we can again define a voter profile (this time the ith row of the table $W(i, j)$ gives such a profile) for each voter, and we need to keep at most B voters with the same profile. Hence, no more than $B(B+1)^B$ voters are present in the instance. Therefore, some brute-force approach can be used to show membership in FPT. ❑

The area of parameterized complexity leaves some freedom regarding the choice of parameterization. The main reason that the standard parameterization (referring to the entity to be minimized, in this case the budget) yields an FPT result is the fact that the parameter is already very big compared to the overall input (e.g., the number of issues n) by the very definition of the problem: Since the money given to one voter will be evenly distributed among the issues and since the cost matrix contains only integer entries, it makes no sense at all to spend less than n dollars on a voter. Hence,

[6] Formally, this means that we have derived a so-called problem kernel for this problem.

the budget should be at least n dollars (assuming that some of the voters must be influenced by The Lobby to achieve their agenda). This obstacle can be sidestepped in two ways:

- by changing the parameterization to B/n, i.e., to the "budget per issue" (see, e.g., Theorem 12);
- by allowing rational numbers as entries in the cost matrix (see, e.g., Theorem 14).

Theorem 12 VB-SM-PLP *and* VB-AM-PLP *(parameterized by the budget per issue) are* W[2]-*complete.*

Proof. We give the details for the case of VB-SM-PLP only; the proof for VB-SM-PLP is analogous but more tedious.[7] W[2]-hardness can be derived from the proof of Theorem 7. To show membership in W[2], we reduce VB-SM-PLP to SMNTMC, which was defined in Section 3. To this end, it suffices to describe how a nondeterministic multi-tape Turing machine can solve such a lobbying problem.

Consider an instance of VB-SM-PLP: a probability matrix $P \in \mathbb{Q}_{[0,1]}^{m\times n}$ with a cost matrix C_P and a budget B. Again, we may assume that the target vector is $\vec{Z} = 1^n$. Moreover, we assume that the target threshold t is fixed. We can identify t with a certain step level for the price functions.

The reducing machine works as follows. From P, C_P, and t, the machine extracts the information $H_{i,j}(d)$, $1 \leq d \leq B$, where $H_{i,j}(d)$ is true if either $p_{i,j} \geq t$ or $c_{i,j}(t) \leq d/n$ (since according to this scenario, the bribery money is evenly distributed across all issues). Note that $H_{i,j}(d)$ captures whether paying d dollars to voter v_i helps to raise the acceptance probability of v_i on referendum r_j above the threshold t. Moreover, for each referendum r_j, the reducing machine computes the minimum number of voters that need to switch their opinion so that majority is reached for that specific referendum; let $s(j)$ denote this threshold for r_j. Since the cost matrix contains integer entries, meaningfully bribing s voters costs at least $s \cdot n$ dollars; only then each referendum will receive at least one dollar per voter. Hence, a referendum with $s(j) > B/n$ yields a NO instance. We can therefore replace any value $s(j) > B/n$ by the value $\lfloor B/n \rfloor + 1$.

The nondeterministic multi-tape Turing machine M we describe next has, in particular, access to $H_{i,j}$ and to $s(j)$. M has $n+1$ working tapes T_j, $0 \leq j \leq n$, all except one of which correspond to issues r_j, $1 \leq j \leq n$. We will use the set of voters, $V = \{v_1, \dots, v_m\}$, as alphabet. The (formal) input tape of M is ignored.

M starts by writing $s(j)$ symbols # onto tape j for each j, $1 \leq j \leq n$. By using parallel writing steps, this needs at most $\lfloor B/n \rfloor + 1$ steps, since $s(j) \leq \lfloor B/n \rfloor + 1$ as argued above.

Second, for each $i \in \{1, \dots, m\}$, M writes k_i symbols v_i from the alphabet V on the zeroth tape, T_0, such that $\sum_{i=1}^{m} k_i \leq B/n$. This is the nondeterministic guessing phase where the amount of bribery money spent on each voter, namely $k_i \cdot n$ for voter v_i, is determined. Note that we can assume that the bribery money is spent in multiples of n, the number of referenda, since spending n dollars on some voter means spending one dollar per issue for that voter.

[7] In particular, the definition of $H_{i,j}$ given below should be generalized, displaying the number of levels by which investing d dollars on voter v_i would raise his or her voting behavior on issue r_j in favor of the The Lobby's agenda.

In the third phase, for each voter v_i that will be bribed, M counts the corresponding amount $k_i \cdot n$ of bribery money and determines (by using $H_{i,j}$) if it is enough to change v_i's opinion regarding the jth issue or if $p_{i,j} \geq t$. If so, the head of M on tape j moves one step to the left. Again, all these head moves are performed in parallel. Hence, the string on the zeroth tape is being processed in at most B/n (parallel) steps.

Finally, it is checked if the left border is reached (again) for all tapes T_j, $j > 0$. This is the case if and only if the guessed bribery was successful. ❑

Theorem 13 VB-AM-PLP *(parameterized by the budget) is in* W[2].

Proof. This can be seen by revisiting the proof of Theorem 12, basically replacing the fraction B/n by the whole budget B. ❑

It is unclear to us whether a corresponding hardness result for VB-AM-PLP is true under the budget parameterization. For the NP-hardness proof of Theorem 7, the number of referenda should not stay constant, so that the relationship between parameterizations is not so clear.

If C_P may contain arbitrary positive rational numbers but we are only allowed to pay whole dollars to the voters, we can also derive hardness results. This scenario might look a bit artificial at first glance, but looking at how prices for gas at a gas station or also how exchange rates between currencies are fixed, it is clear that the fact that rounding must come into play in the process of payment at some stage, although the prices per item might be given not in whole currency units. Similarly, in some countries, grocers might ask for 1.99 dollars for some product, although there are no coins smaller than 5 cents available.[8]

Hence, consider the generalization of our problems where the cost matrix may contain arbitrary positive rational numbers. The corresponding problems are prefixed by a $\mathbb{Q}$.

Theorem 14 $\mathbb{Q}$-VB-SM-PLP *and* $\mathbb{Q}$-VB-AM-PLP *(parameterized by the budget) are* W[2]*-complete.*

Proof. For proving W[2]-hardness, the NP-hardness proof of Theorem 7 has to be modified at three points: C_P will no longer contain only zeros and ones, but rather the numbers zero and $1/n$, so that then one unit of money will be spent per voter. The budget B of the derived instance will then be equal to the number b of voters in the given OL-instance.

For showing membership in W[2], we can use the proof of Theorem 13, keeping in mind that the only point where the modification comes into play is when computing $H_{i,j}(d)$, which can be done by the reducing machine. ❑

[8] An example is the Hungarian currency.

5.2 Probabilistic Lobbying with Issue Weighting

Recall from Theorem 8 that MB-SM-PLP-WIW, MB-AM-PLP-WIW, IB-SM-PLP-WIW, and IB-AM-PLP-WIW are NP-complete. We now show that each of these problems is fixed-parameter tractable when parameterized by the budget.

Theorem 15 *The problems* MB-SM-PLP-WIW, MB-AM-PLP-WIW, IB-SM-PLP-WIW, *and* IB-AM-PLP-WIW *(parameterized by the budget) are in* FPT.

Proof. Since the four unweighted variants are in P, we can compute the number of dollars to be spent to win referendum r_j in polynomial time in each case. Now re-interpret the given X-Y-PLP-WIW instance, where $\mathrm{X} \in \{\mathrm{MB}, \mathrm{IB}\}$ and $\mathrm{Y} \in \{\mathrm{SM}, \mathrm{AM}\}$, as a KNAPSACK instance: Every issue r_j is an object o_j with weight d_j and profit p_j, both set to be the same as weight w_j of issue r_j. Let the KNAPSACK bound be the total number B of dollars allowed to be spent. Now use the pseudo-polynomial algorithm (see [16]) to solve KNAPSACK in time $\mathcal{O}(n2^{|B|})$, where $|B|$ denotes the length of the encoding of B. ❑

Voter bribery with issue weighting keeps its complexity status for both evaluation criteria.

Since it is not hard to incorporate issue weights into brute-force computations, we have the following corollary from Theorems 10 and 11.

Corollary 16
1. VB-SM-PLP-WIW *and* VB-AM-PLP-WIW *(parameterized by the budget and by the discretization level) are in* FPT.
2. VB-SM-PLP-WIW *(parameterized by the budget) is in* FPT.

Theorem 17 VB-SM-PLP-WIW *and* VB-AM-PLP-WIW *(parameterized by the budget per issue B/n and by the objective O) are* W[2]*-complete.*

Proof. By the proof of Theorem 7, VB-SM-PLP (parameterized by the budget per issue) is W[2]-hard, since we can consider unit weights and $O = 1$ as the objective. Since VB-SM-PLP is the special case of VB-SM-PLP-WIW where all the issues have unit weight, VB-SM-PLP-WIW is W[2]-hard as well. By an analogous argument, VB-AM-PLP-WIW is W[2]-hard, too.

Membership in W[2] is a bit more tricky than in the unweighted case from Theorem 12. In the following, we indicate only the necessary modifications:

- The reducing machine calculates the difference O' between the target weight and the sum of the weights of the referenda that are already won.
- The reducing machine can replace issue weights bigger than O' with $O' + 1$.
- For each referendum that is not already won, the reducing machine introduces a special letter r_i to be used on the zeroth tape.

- The Turing machine that has been built at the very beginning also guesses at most B referenda that (additionally) should be won. (Note that influencing any issue costs at least one dollar.) Then, the Turing machine will spend $\mathcal{O}(f(B,O))$ time to calculate if winning those guessed referenda $r_{i,1},\ldots,r_{i,b}$, $b \leq B$, would be sufficient to get beyond the threshold O. Only if sufficiency is guaranteed, the Turing machine continues working.

- The Turing machine will then continue to work as described in the proof of Theorem 12.

- At the very end, the Turing machine will verify in at most B steps if all referenda guessed in the very beginning have been won.

W[2]-completeness of VB-AM-PLP-WIW (parameterized by the budget per issue B/n and by the objective O) can be proven by an analogous argument. ❑

Theorem 18 VB-AM-PLP-WIW *(parameterized by the budget B and by the objective O) is in* W[2].

Proof. This follows by the arguments for membership in W[2] given in the proof of Theorem 17 (which itself is built on top of and modifies some parts of the proof of Theorem 12), basically replacing the fraction B/n by the whole budget B. ❑

Note that it is quite tempting to try to avoid the weight calculations within the Turing machine in the proof of Theorem 17, letting the reducing machine do this job. However, this seems to necessitate coding the winning situations into the state set of the Turing machine, leading to a possibly exponential size of this Turing machine (measured in the overall input size of the lobbying scenario). As it is unclear to us whether this obstacle can be avoided, we leave it open if membership in W[2] can be shown using only the parameter B/n (and not the objective O). However, we can state the following.

Corollary 19 VB-SM-PLP-WIW *and* VB-AM-PLP-WIW *(parameterized by the budget per issue B/n) are* W[2]*-hard.*

Similar modifications show:

Corollary 20 $\mathbb{Q}$-VB-SM-PLP-WIW *and* $\mathbb{Q}$-VB-AM-PLP-WIW *(parameterized by the budget and by the objective value O) are* W[2]*-complete.*

Corollary 21 $\mathbb{Q}$-VB-SM-PLP-WIW *and* $\mathbb{Q}$-VB-AM-PLP-WIW *(parameterized by the budget) are* W[2]*-hard.*

6 Approximability

As seen in Tables 1 and 2, many problem variants of probabilistic lobbying are NP-complete. Hence, it is interesting to study them not only from the viewpoint of parameterized complexity, but also from the viewpoint of approximability.

The budget constraint on the bribery problems studied so far gives rise to natural minimization problems: Try to minimize the amount spent on bribing. For clarity, let us denote these minimization problems by prefixing the problem name with MIN, leading to, e.g., MIN-OL.

6.1 Voter Bribery is Hard to Approximate

The already-mentioned reduction of Christian et al. [5] (that proved that OL is W[2]-hard) is parameter-preserving (regarding the budget). It further has the property that a possible solution found in the OL instance can be re-interpreted as a solution to the DOMINATING SET instance that the reduction started with, and the OL solution and the DOMINATING SET solution are of the same size. This in particular means that inapproximability results for DOMINATING SET transfer to inapproximability results for OL. Similar observations are true for the interrelation of SET COVER and DOMINATING SET, as well as for OL and VB-SM-PLP-WIW (or VB-AM-PLP-WIW).

The known inapproximability results [3,19] for SET COVER hence give the following result (see also Footnote 4 in [21]).

Theorem 22 *There is a constant $c > 0$ such that* MIN-OL *is not approximable within factor $c \cdot \log n$ unless* NP $\subset$ DTIME$(n^{\log\log n})$*, where n denotes the number of issues.*

Since OL can be viewed as a special case of both VB-Y-PLP and VB-Y-PLP-WIW for $\mathrm{Y} \in \{\mathrm{SM}, \mathrm{AM}\}$, we have the following corollary.

Corollary 23 *For* $\mathrm{Y} \in \{\mathrm{SM}, \mathrm{AM}\}$*, there is a constant $c_{\mathrm{Y}} > 0$ such that both* MIN-VB-Y-PLP *and* MIN-VB-Y-PLP-WIW *are not approximable within factor $c_{\mathrm{Y}} \cdot \log n$ unless* NP $\subset$ DTIME$(n^{\log\log n})$*, where n denotes the number of issues.*

Proof. How to interpret an instance of OL as a VB-Y-PLP-instance, $\mathrm{Y} \in \{\mathrm{SM}, \mathrm{AM}\}$, is given in detail in the proof of Theorem 7. The relation $B = n \cdot b$ between the budget B and the number of voters b holds both for optimum and for approximate solutions. Hence, the n is canceled out when looking at the approximation ratio. ❑

Conversely, a logarithmic-factor approximation can be given for the minimum-budget versions of all our problems, as we will show now. We first discuss the relation to the well-known SET COVER problem, sketching a tempting, yet flawed reduction and pointing out its pitfalls. Avoiding these pitfalls, we then give an approximation algorithm for MIN-VB-AM-PLP. Moreover, we define the notion of cover number, which allows to state inapproximability results for MIN-VB-AM-PLP. Similar results hold for MIN-VB-SM-PLP, the constructions being sketched at the end of the section.

Voter bribery problems are closely related to set cover problems, in particular in the average-majority scenario, so that we should be able to carry over approximability ideas from that area. The intuitive translation of a MIN-VB-AM-PLP instance into a SET COVER instance is as follows: The universe of the derived SET COVER instance should be the set of issues, and the sets (in the SET COVER instance) are formed by considering the sets of issues that could be influenced (by changing a voter's opinion) through bribery of a specific voter. Namely, when we pay voter v a specific amount of money, say d dollars, he or she will credit d/n dollars to each issue and possibly change v's opinion (or at least raise v's acceptance probability to some "higher level"). The weights associated with the sets of issues correspond to the bribery costs that are (minimally) incurred to lift the issues in the set to some "higher level." There are four differences to classical set covering problems:

1. We cannot neglect the voter who has been bribed, so different voters (with different bribing costs) may be associated with the same set of issues.

2. The sets associated with one voter are not independent. For each voter, the sets of issues that can be influenced by bribing that voter are linearly ordered by set inclusion. Moreover, when bribing a specific voter, we have to first influence the "smaller sets" (which might be expensive) before possibly influencing the "larger ones"; so, weights are attached to set differences, rather than to sets.

3. A *cover number* $c(r_j)$ is associated with each issue r_j, indicating by how many levels voters must raise their acceptance probabilities in order to arrive at average majority for r_j. The cover numbers can be computed beforehand for a given instance. Then, we can also associate cover numbers with sets of issues (by summation), which finally leads to the cover number $N = \sum_{j=1}^{n} c(r_j)$ of the whole instance.

4. The money paid "per issue" might not have been sufficient for influencing a certain issue up to a certain level, but it is not "lost"; rather, it would make the next bribery step cheaper, hence (again) changing weights in the set cover interpretation.

To understand these connections better, let us have another look at our running example (under voter bribery with average-majority evaluation, i.e., MIN-VB-AM-PLP), assuming an all-ones target vector. If we paid 30 dollars to voter v_1, he or she would credit 10 dollars to each issue, which would raise his or her acceptance probability for the second issue from .3 to .4; no other issue level is changed. Hence, this would correspond to a set containing only r_2 with weight 30. Note that by this bribery, the costs for raising the acceptance probability of voter v_1 to the next level would be lowered for the other two issues. For example, spending 15 more dollars on v_1 would raise r_3 from .5 to .6, since all in all 45 dollars have been spent on voter v_1, which means 15 dollars per issue. If the threshold is 60% in that example, then the first issue is already accepted (as desired by The Lobby), but the second issue has gone up from .5 to .6 on average, which means that we have to raise either the acceptance probability of one voter by two levels (for example, by paying 210 dollars to voter v_1), or we have to raise the acceptance probability of each voter by one level (by paying 30 dollars to voter v_1 and another 30 dollars to voter v_2). This can be expressed by saying that the first issue has a cover number of zero, and the second has a cover number of two.

When we interpret an OL instance as a VB-AM-PLP instance, the cover number of the resulting instance equals the number of issues, assuming that the votes for all issues need amendment. Thus we have the following corollary:

Corollary 24 *There is a constant $c > 0$ such that* MIN-VB-AM-PLP *is not approximable within factor $c \cdot \log N$ unless* $\mathrm{NP} \subset \mathrm{DTIME}(N^{\log\log N})$*, where N is the cover number of the given instance. A fortiori, the same statement holds for* MIN-VB-AM-PLP-WIW.

Let H denote the harmonic sum function, i.e., $H(r) = \sum_{i=1}^{r} 1/i$. It is well known that $H(r) = O(\log r)$. More precisely, it is known that

$$\lfloor \ln r \rfloor \leq H(r) \leq \lfloor \ln r \rfloor + 1.$$

We now show the following theorem.

Theorem 25 MIN-VB-AM-PLP *can be approximated within a factor of $\ln(N) + 1$, where N is the cover number of the given instance.*

Proof. Consider the greedy algorithm shown in Figure 1, where t is the given threshold and we assume, without loss of generality, that The Lobby has the all-ones target vector. Note that the cover numbers (per referendum) can be computed from the cost matrix C_P and the threshold t before calling the algorithm the very first time.

Observe that our greedy algorithm influences voters only via raising their acceptance probabilities by only one level, so that the amount d_v possibly spent on voter v in Step 3 of the algorithm actually corresponds to a set of referenda; we do not have to consider multiplicities of issues (raised over several levels) here.

Let $S_1, \ldots, S_\ell$ be the sequence of sets of referenda picked by the greedy bribery algorithm, along with the sequence $v_1, \ldots, v_\ell$ of voters and the sequence $d_1, \ldots, d_\ell$ of bribery dollars spent this way. Let $R_1 = R, \ldots, R_\ell, R_{\ell+1} = \emptyset$ be the corresponding sequence of sets of referenda, with the accordingly modified cover numbers c_i. Let $j(r,k)$ denote the index of the set in the sequence influencing referendum r the kth time with $k \leq c(r)$, i.e., $r \in S_{j(r,k)}$ and $|\{i < j(r,k) \mid r \in S_i\}| = k - 1$. To cover r the kth time, we have to pay $\chi(r,k) = d_{j(r,k)}/|S_{j(r,k)}|$ dollars. The greedy algorithm will incur a cost of $\chi_{greedy} = \sum_{r \in R} \sum_{k=1}^{c(r)} \chi(r,k)$ in total.

An alternative view on the greedy algorithm is from the perspective of the referenda: By running the algorithm, we implicitly define a sequence $r_1, \ldots, r_N$ of referenda, where $N = c(R) = \sum_{r \in R} c(r)$ is the cover number of the original instance, such that $S_1 = \{r_1, \ldots, r_{|S_1|}\}$, $S_2 = \{r_{|S_1|+1}, \ldots, r_{|S_1|+|S_2|}\}$, etc. Ties (how to list elements within S_i) are broken arbitrarily. This (implicitly) defines two functions $L, R : \{1, \ldots, \ell\} \to \{1, \ldots, N\}$ such that $S_i = \{r_{L(i)}, \ldots, r_{R(i)}\}$. Slightly abusing notation, we can associate a cost $\chi'(r_i)$ with each element in the sequence (keeping in mind the multiplicities of covering implied by the sequence), so that $\chi_{greedy} = \sum_{i=1}^{N} \chi'(r_i)$. Note that $d_i = \sum_{L(i) \leq r \leq R(i)} \chi'(x_r)$.

Consider r_j with $L(i) \leq j \leq R(i)$. The current referendum set R_i has cover number $N - L(i) + 1$, i.e., of at least $N - j + 1$. Let χ_{opt} be the cost of an optimum bribery strategy $\mathscr{C}^*$ of the original universe. Of course, this also yields a cover of the referendum set R_i with cost at most χ_{opt}.

Input: A probability matrix $P \in \mathbb{Q}_{[0,1]}^{m \times n}$ (implicitly specifying a set V of m voters and a set R of n referenda), a cost matrix C_P, and n cover numbers $c(r_1), \ldots, c(r_n) \in \mathbb{N}$.

1. Delete referenda that are already won (indicated by $c(r_j) = 0$), and modify R and C_P accordingly.
2. If $R = \emptyset$ then output the amount spent on bribing so far and STOP.
3. For each voter v, compute the least amount of money, d_v, that could raise any level in C_P. Let n_v be the number of referenda whose levels are raised when spending d_v dollars on voter v.
4. Bribe voter v such that d_v/n_v is minimum.
5. Modify C_P by subtracting d_v/n from each amount listed for voter v.
6. Modify c by subtracting one from $c(r)$ for those referenda $r \in R$ influenced by this bribery action.
7. Recurse.

Figure 1: Greedy approximation algorithm for MIN-VB-AM-PLP in Theorem 25

The average cost per element (taking into account multiplicities as given by the cover numbers) is $\chi_{opt}/c(R_i)$. (So, whether or not some new levels are obtained through bribery does not really matter here, as long as the threshold is not exceeded.)

$\mathscr{C}^*$ can be described by a sequence of sets of referenda $C_1, \ldots, C_q$, with corresponding voters $z_1, \ldots, z_q$ and dollars $d_1^*, \ldots, d_q^*$ spent. Hence, $\chi_{opt} = \sum_{\kappa=1}^{q} d_\kappa^*$. With each bribery step we associate the cost factor $d_\kappa^*/|C_\kappa|$, for each issue r contained in C_κ. $\mathscr{C}^*$ could be also viewed as a bribery strategy for R_i. By the pigeon hole principle, there is a referendum r in R_i (to be influenced the kth time) with cost factor at most $d_\kappa^*/|C_\kappa \cap R_i| \leq \chi_{opt}/c(R_i)$, where κ is the index such that C_κ contains r for the kth time in $\mathscr{C}^*$ (usually, the cost would be smaller, since part of the bribery has already been paid before). Since (S_i, v_i) was picked to minimize $d_i/|S_i|$, we find $d_i/|S_i| \leq d_\kappa^*/|C_\kappa \cap R_i| \leq \chi_{opt}/c(R_i)$.

We conclude that

$$\chi'(r_j) \leq \frac{\chi_{opt}}{c(R_i)} = \frac{\chi_{opt}}{N - L(i) + 1} \leq \frac{\chi_{opt}}{N - j + 1}.$$

Hence,

$$\chi_{greedy} = \sum_{j=1}^{N} \chi'(r_j) \leq \sum_{j=1}^{N} \frac{\chi_{opt}}{N - j + 1} = H(N)\chi_{opt} \leq (\ln(N) + 1)\chi_{opt},$$

which completes the proof. ❑

In the strict-majority scenario, cover numbers would have a different meaning—we thus call them *strict cover numbers*: For each referendum, the corresponding strict cover number tells in advance how many voters have to change their opinions (bringing them individually over the given threshold t) to accept this referendum. Again, the strict cover number of a problem instance is the sum of the strict cover numbers of all given referenda.

The corresponding greedy algorithm would therefore choose to influence voter v_i (with d_i dollars) in the ith loop so that v_i changes his or her opinion on some referendum r_j such that $d_i/|\rho_j|$ is minimized.[9]

We can now read the approximation estimate proof given for the average-majority scenario nearly literally as before, by re-interpreting the formulation "influencing referendum r" meaning now a complete change of opinion for a certain voter (not just gaining one level somehow). This establishes the following result.

Theorem 26 MIN-VB-SM-PLP *can be approximated within a factor of* $\ln(N)+1$, *where N is the strict cover number of the given instance.*

Note that this result is in some sense stronger than Theorem 25 (which refers to the average-majority scenario), since the cover number of an instance could be larger than the strict cover number.

This approximation result is complemented by a corresponding hardness result.

Theorem 27 *There is a constant $c > 0$ such that* MIN-VB-SM-PLP *is not approximable within factor $c \cdot \log N$ unless* $\mathrm{NP} \subset \mathrm{DTIME}(N^{\log\log N})$, *where N is the strict cover number of the given instance. A fortiori, the same statement holds for* MIN-VB-SM-PLP-WIW.

Unfortunately, those greedy algorithms do not (immediately) transfer to the case when issue weights are allowed. These weights might also influence the quality of approximation, but a simplistic greedy algorithm might result in covering the "wrong" arguments. Also, the proof of the approximation factor given above will not carry over, since we need as one of the proof's basic ingredients that an optimum solution can be interpreted as a partial one at some point. Those problems tend to have a different flavor.

6.2 Polynomial-Time Approximation Schemes

Those problems for which we obtained FPT results in the case of issue weights actually enjoy a polynomial-time approximation scheme (PTAS).[10] The proof of Theorem 15 can be easily turned into proving a PTAS, since that result was obtained by transferring pseudo-polynomial time algorithms.

[9] Possibly, there is a whole set ρ_j of referenda influenced this way.

[10] A polynomial-time approximation scheme is an algorithm that for each pair (x,ε), where x is an instance of an optimization problem and $\varepsilon > 0$ is a rational constant, runs in time polynomial in $|x|$ and outputs an approximate solution for x within a factor of ε.

Theorem 28 *Each of the problems* MIN-MB-SM-PLP-WIW, MIN-MB-AM-PLP-WIW, MIN-IB-SM-PLP-WIW, *and* MIN-IB-AM-PLP-WIW *admits a PTAS.*

As a final remark, the exact version of microbribery also admits an FPT result, but this cannot be interpreted as an approximation result, since the entity that should be minimized has to be hit exactly (otherwise, we have polynomial time).

7 Conclusions

We have studied six lobbying scenarios in a probabilistic setting, both with and without issue weights. Among the problems studied and their variants, we identified those that can be solved in polynomial time, those that are NP-complete yet fixed-parameter tractable, and those that are hard (namely, W[2]-complete) in terms of their parameterized complexity with suitable parameters. We also investigated the approximability of hard probabilistic lobbying problems (without issue weights) and obtained both approximation and inapproximability results.

An interesting direction for future work would be to study the parameterized complexity results under different parameterizations. We would also like to investigate the open question of whether one can find logarithmic-factor approximations for voter bribery with issue weights.

From the viewpoint of parameterized complexity, it would be quite interesting to solve the question of whether or not certain versions of probabilistic lobbying problems involving issue weights belong to W[2] (see the question marks in Table 2). In fact, we did not see how to put these problems in any level of the W-hierarchy. Problems that involve numbers seem to have a particular flavor that makes them hard to tackle with these techniques, but we suggest this to be the subject of further studies. Note that all probabilistic lobbying problems that have been connected with W[2] somehow in this paper are in fact strongly NP-hard, i.e., their hardness does not depend on whether the numbers in their inputs are encoded in unary or in binary. We suspect that this adds to the difficulty when dealing with these problems from a parameterized perspective.

References

[1] M. Baye, D. Kovenock, and C. de Vries. Rigging the lobbying process: An application of the all-pay auction. *The American Economic Review*, 83(1):289–294, 1993.

[2] M. Baye, D. Kovenock, and C. de Vries. The all-pay auction with complete information. *Economic Theory*, 8(2):291–305, 1996.

[3] M. Bellare, S. Goldwasser, C. Lund, and A. Russell. Efficient probabilistically checkable proofs and applications to approximations. In *Proceedings of the 25th ACM Symposium on Theory of Computing*, pages 294–304. ACM Press, 1993.

[4] M. Cesati. The Turing way to parameterized complexity. *Journal of Computer and System Sciences*, 67:654–685, 2003.

[5] R. Christian, M. Fellows, F. Rosamond, and A. Slinko. On complexity of lobbying in multiple referenda. *Review of Economic Design*, 11(3):217–224, 2007.

[6] C. Crombez. Information, lobbying and the legislative process in the European Union. *European Union Politics*, 3(1):7–32, 2002.

[7] R. Downey and M. Fellows. *Parameterized Complexity*. Springer-Verlag, 1999.

[8] G. Erdélyi, H. Fernau, J. Goldsmith, N. Mattei, D. Raible, and J. Rothe. The complexity of probabilistic lobbying. In *Proceedings of the 1st International Conference on Algorithmic Decision Theory*, pages 86–97. Springer-Verlag *Lecture Notes in Artificial Intelligence #5783*, October 2009.

[9] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. The complexity of bribery in elections. *Journal of Artificial Intelligence Research*, 35:485–532, 2009.

[10] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Llull and Copeland voting broadly resist bribery and control. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, pages 724–730. AAAI Press, July 2007.

[11] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Copeland voting fully resists constructive control. In *Proceedings of the 4th International Conference on Algorithmic Aspects in Information and Management*, pages 165–176. Springer-Verlag *Lecture Notes in Computer Science #5034*, June 2008.

[12] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Llull and Copeland voting computationally resist bribery and constructive control. *Journal of Artificial Intelligence Research*, 35:275–341, 2009.

[13] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. A richer understanding of the complexity of election systems. In S. Ravi and S. Shukla, editors, *Fundamental Problems in Computing: Essays in Honor of Professor Daniel J. Rosenkrantz*, chapter 14, pages 375–406. Springer, 2009.

[14] J. Flum and M. Grohe. *Parameterized Complexity Theory*. EATCS Texts in Theoretical Computer Science. Springer-Verlag, 2006.

[15] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

[16] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems*. Springer-Verlag, Berlin, Heidelberg, New York, 2004.

[17] C. Lindner and J. Rothe. Fixed-parameter tractability and parameterized complexity, applied to problems from computational social choice. In A. Holder, editor, *Mathematical Programming Glossary*. INFORMS Computing Society, Oct. 2008.

[18] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.

[19] R. Raz and S. Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In *Proceedings of the 29th ACM Symposium on Theory of Computing*, pages 475–484. ACM Press, 1997.

[20] J. Reinganum. A formal theory of lobbying behaviour. *Optimal Control Applications and Methods*, 4(4):71–84, 1983.

[21] T. Sandholm, S. Suri, A. Gilpin, and D. Levine. Winner determination in combinatorial auction generalizations. In *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 69–76. ACM Press, July 2002.

[22] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944.